



⑮ **BUNDESREPUBLIK  
DEUTSCHLAND**



**DEUTSCHES  
PATENTAMT**

⑫ **Offenlegungsschrift**  
⑩ **DE 195 22 527 A 1**

⑤ Int. Cl.®:  
**G 06 K 7/00**  
G 06 K 19/07

⑲ Aktenzeichen: 195 22 527.9  
⑳ Anmeldetag: 23. 8. 95  
㉓ Offenlegungstag: 2. 1. 97

**DE 195 22 527 A 1**

⑦① Anmelder:

International Business Machines Corp., Armonk,  
N.Y., US

⑦④ Vertreter:

Schäfer, W., Dipl.-Ing., Pat.-Anw., 70176 Stuttgart

⑦② Erfinder:

Rindtorff, Klaus, Dipl.-Inform., 71093 Weil im  
Schönbuch, DE; Bublitz, Hermann, Dipl.-Ing., 71034  
Böblingen, DE

⑤⑥ Entgegenhaltungen:

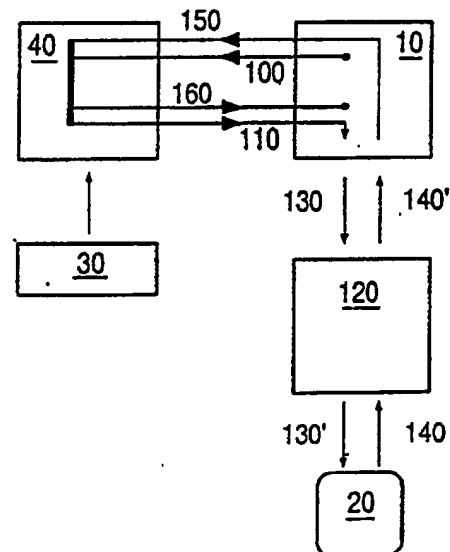
DE 1 95 08 940 A1  
US 52 78 312

BEUTELSPACHER, A, KERSTEN, PFAU: Chipkarten  
als Sicherheitswerkzeug, Springer Verlag 1991,  
ISBN: 3-540-54140-3, S. 78-85;

Prüfungsantrag gem. § 44 PatG ist gestellt

⑤④ Verfahren zur Vereinfachung der Kommunikation mit Chipkarten

⑤⑦ Eine erfindungsgemäße Vorrichtung zur Kommunikation einer Anwendung (10) mit einer Chipkarte (20) weist mindestens ein Anwendungsdatenverzeichnis (30) - oder auch Wörterbuch genannt - zur Aufnahme von Informationen über anwendungsspezifische Daten der Anwendung (10), und mindestens ein Chipkartendialogmodul (40) - oder auch Agent genannt - für die Generierung von Kommandos mit Hilfe des Anwendungsdatenverzeichnisses (30) für eine Schnittstelle zur Chipkarte (20) auf, wobei das Chipkartendialogmodul (40) kartenspezifische Daten über die Chipkarte (20) enthält. Von der Anwendung (10) wird eine Anfrage (100) zur Kommunikation mit der Chipkarte (20) an das für die Chipkarte (20) spezifizierte Chipkartendialogmodul (40) gestellt. Das Chipkartendialogmodul (40) erzeugt auf die Anfrage (100) hin mindestens ein Kommando (110), das zu einer Kommunikation mit der Chipkarte (20) erforderlich ist. Das Chipkartendialogmodul (40) macht sich hierfür anwendungsspezifische Informationen, die in dem Anwendungsdatenverzeichnis (30) gespeichert sind, zugänglich.



**DE 195 22 527 A 1**

## Beschreibung

## Gebiet der Erfindung

Die Erfindung betrifft eine Vorrichtung und ein Verfahren zur Kommunikation einer Anwendung mit einer Chipkarte, sowie eine Verwendung in Lese-/Schreibgeräten für Chipkarten.

## Stand der Technik

Als Datenträgerkarten oder Chipkarten werden heute tragbare, zumeist kleine, in etwa in Scheckkartenformat sich befindliche, Karten, vorzugsweise aus Kunststoff oder Metall, mit einem darin integrierten elektronischen Chip bezeichnet. Im Gegensatz zu einfachen Speicherkarten (die als Memory Chip-Cards oder Memory-Cards bekannt sind) besitzen intelligente Datenträgerkarten (auch multifunktionale Chipkarten, oder Smart-Cards genannt) neben einem Speicherbereich einen eigenen Prozessor zur Kontrolle der auf dem Chip der Datenträgerkarte gespeicherten Daten. Dies erlaubt einen besseren Schutz der Daten und führt zu einer verbesserten Funktionalität der Chipkarten. Einfache Speicherkarten erlauben im allgemeinen nur ein Schreiben und Lesen der Daten. Intelligente Datenträgerkarten verfügen darüber hinaus noch über Funktionen zur Strukturierung der Daten, zur Lokalisierung der Daten, zur Verwaltung der Daten und zum Schutz der Daten. Die Schnittstelle einer intelligenten Chipkarte, und damit auch die erforderliche Programmierung (z. B. von Kommandosequenzen), um Daten von einer Chipkarte zu lesen, ist daher wesentlich komplexer als die von Speicherkarten.

Der vor allem wegen der erhöhten Fälschungssicherheit zunehmende Einsatz von Chipkarten erstreckt sich auf zahlreiche Anwendungsgebiete. Anwendungen mit Chipkarten, also Anwendungen, für die eine Kommunikation eines beliebigen Gerätes mit einer Chipkarte erforderlich ist, können beispielsweise das bargeldlose Bezahlen, Identifikation des Chipkarteninhabers, die Speicherung von Daten oder dergleichen sein. Eine Anwendung besteht dabei aus internen Anwendungsteilen auf der Chipkarte und externen Anwendungsteilen in entsprechenden Geräten, wie z. B. Geldautomaten, PC's oder speziellen Terminals. Allgemein stellen interne Anwendungsteile alle Daten und Programme dar, die auf der Chipkarte selbst gespeichert werden, während die externen Anwendungsteile alle Daten und Programme außerhalb der Chipkarte darstellen. Die Programmierung dieser externen Anwendungsteile obliegt meist den Programmierern der Geräte, mit denen eine Chipkarte zum Einsatz kommen soll. Häufig existiert in diesen Geräten bereits eine komplexe Software-Plattform. Um hier Anwendungen mit Chipkarten zu integrieren, sind umfangreiche Kenntnisse über die Struktur der dort gespeicherten Daten und die Schnittstelle der Chipkarte erforderlich.

Die meisten verwandten Chipkarten unterscheiden sich von Hersteller zu Hersteller deutlich voneinander. Viele der Chipkarten implementieren eine Teilmenge des ISO Standards 7816 und verfügen zusätzlich über einige Sonderfunktionen. Dazu kommt in der Praxis die erforderliche Kenntnis der Details der zu implementierenden internen und externen Anwendungsteile.

Um einen Zugriff auf die auf Chipkarten gespeicherten Daten zu ermöglichen, erfolgt zumeist eine feste Programmierung (Kodierung) von Kommandos. Dies bedeutet jedoch eine erhöhte Starrheit der Anwendung und eine schlechte Wartbarkeit.

Beim Einsatz von Chipkarten wird im allgemeinen eine Verschlüsselung der Daten oder der Kommandos zur Authentisierung der Chipkarte oder der, zur Chipkarte externen Welt verwendet. Meist wird dabei ein symmetrischer Verschlüsselungsalgorithmus, wie der DES (digital encryption standard), oder ein asymmetrischer Verschlüsselungsalgorithmus, wie z. B. ein Public-Key Algorithmus, benutzt. Zur Authentisierung des Chipkarteninhabers kommt derzeit fast ausschließlich eine Identifikationsnummer PIN (Personal Identification Number) zum Einsatz. Neben Geräten, wie Lese-/Schreibgeräten für Chipkarten, sind daher insbesondere noch weitere Elemente, wie die Authentisierungselemente zur Verschlüsselung und zur Eingabe der PIN, für eine Anwendung mit Chipkarten zu berücksichtigen und können an einer solchen Anwendung beteiligt sein.

Eine Kommunikation der, durch Hardware oder Software realisierten, Elemente und Geräte mit den externen Anwendungsteilen geschieht durch die Verwendung von Programmierschnittstellen. Für eine Chipkarte ist eine einfache Schnittstelle jedoch nicht befriedigend. Durch die Komplexität der Datenstrukturen und der erforderlichen Kommandos reicht es nicht aus, eine Einbettung der Kommandos in ein anderes Protokoll oder eine höhere Programmiersprache zu verwenden. Dies setzt auch, für anwendungsspezifische Daten, die Kenntnis über interne Strukturen der Chipkarte und damit auch über ihre Erreichbarkeit und Zugreifbarkeit durch die Anwendung über die Schnittstelle zur Chipkarte hin, voraus.

## Zusammenfassung der Erfindung

Es ist Aufgabe der Erfindung, eine Vereinfachung der Schnittstelle zur Kommunikation mit einer Chipkarte zu schaffen. Die Aufgabe der Erfindung wird entsprechend der unabhängigen Ansprüche gelöst.

Eine erfindungsgemäße Vorrichtung zur Kommunikation einer Anwendung mit einer Chipkarte weist mindestens ein Anwendungsdatenverzeichnis — oder auch Wörterbuch genannt — zur Aufnahme von Informationen über anwendungsspezifische Daten der Anwendung, und mindestens ein Chipkartendialogmodul — auch Agent genannt — für die Generierung von Kommandos mit Hilfe des Anwendungsdatenverzeichnisses für eine Schnittstelle zur Chipkarte, wobei das Chipkartendialogmodul kartenspezifische Daten über die Chipkarte enthält.

Von der Anwendung wird eine Anfrage zur Kommunikation mit der Chipkarte an das für die Chipkarte

spezifiziertes Chipkartendialogmodul gestellt. Das Chipkartendialogmodul erzeugt auf die Anfrage hin mindestens ein Kommando, das zu einer Kommunikation mit der Chipkarte erforderlich ist. Das Chipkartendialogmodul macht sich hierfür anwendungsspezifische Informationen, die in dem Anwendungsdatenverzeichnis gespeichert sind, zugänglich.

Um die Implementierung von Anwendungen mit unterschiedlichen Typen von Chipkarten zu erleichtern, ist es erforderlich, eine möglichst einheitliche Schnittstelle der Anwendungen zur Chipkarte zu bekommen. Erfindungsgemäß erfolgt eine Trennung zwischen anwendungsspezifischen Daten und kartenspezifischen Daten. Anwendungsspezifische Daten sind solche Daten, die Informationen über Art, Lokalität, Umfang und Zugriffsmethoden für auf einer Chipkarte gespeicherte Daten enthalten, sowie die auf der Chipkarte gespeicherten Daten selbst. Kartenspezifische Daten dagegen stellen solche Daten dar, die Informationen über die erforderlichen Kommandos und das Protokoll einer Chipkarte zum Zugriff auf die dort gespeicherten Daten geben.

Die Trennung zwischen anwendungsspezifischen Daten und kartenspezifischen Daten ermöglicht es, daß ein und dieselbe Anwendung mit verschiedenen Chipkartentypen realisiert werden kann. Dies führt zu einer wesentlichen Vereinfachung der Schnittstelle zu Chipkarten und verbessert andererseits die Trennung zwischen internen Anwendungsteilen auf der Chipkarte und von der Chipkarte getrennten, externen Anwendungsteilen. Eine flexible Anpassung an neue Anwendungen und Chipkartentypen wird so unterstützt.

Die hier vorgestellte Lösung erleichtert sowohl die Integration von Chipkarten in existenten Anwendungen, als auch die Implementierung neuer Anwendungen. Durch die Trennung von anwendungsspezifischen und kartenspezifischen Daten werden die für Anwendungen erforderlichen Kenntnisse minimiert und ihre Verwendung vereinfacht. Die Wartbarkeit der internen als auch externen Anwendungsteile wird deutlich verbessert. Die Anwendung der Chipkarte wird von kartenspezifischen Aspekten wie Kommandos, Protokollen und Datenstrukturen entlastet. Gleichzeitig wird neben einer synchronen auch eine asynchrone Betriebsart ermöglicht.

Die Erfindung findet vorzugsweise Anwendung in Lese-/Schreibgeräten für Chipkarten im weitesten Sinne, das heißt in PC's oder anderen Geräten die eine Kommunikation mit Chipkarten koordinieren, steuern und mittelbar oder unmittelbar durchführen.

Weitere, vorteilhafte Ausführungen der Erfindung finden sich in den Unteransprüchen.

#### Beschreibung der Zeichnungen

Zur näheren Erläuterung der Erfindung sind im folgenden Ausführungsbeispiele mit Bezugnahme auf die Zeichnungen beschrieben. Es zeigen:

Fig. 1 eine schematische Darstellung der erfindungsgemäßen Kommunikation einer Anwendung mit einer Chipkarte,

Fig. 2 die Verwendung mehrerer Wörterbücher und/oder mehrerer Agenten für eine Vielzahl verschiedener Anwendungen und Typen von Chipkarten,

Fig. 3 eine weitere Ausführungsform die eine Reduzierung des Overheads bei der Kommunikation mit der Chipkarte ermöglicht.

#### Beschreibung der Erfindung

Fig. 1 zeigt eine schematische Darstellung der erfindungsgemäßen Kommunikation einer Anwendung 10 mit einer Chipkarte 20. Die Anwendung 10 befindet sich, getrennt von der Chipkarte 20, beispielsweise in einem Lese-/Schreibgerät, einem Computer oder einer sonstigen, zur Kommunikation mit der Chipkarte 20 fähigen Umgebung. Ein Anwendungsdatenverzeichnis, das sogenannte Wörterbuch 30 (application dictionary), dient zur Aufnahme von anwendungsspezifischen Daten. Ein Chipkartendialogmodul, der sogenannte Agent 40 (smart card agent oder smart card interpreter), generiert die erforderlichen Kommandos für die Schnittstelle der Chipkarte 20.

Das Wörterbuch 30 enthält Informationen über Art, Lokalität, Umfang und Zugriffsmethoden von auf einer Chipkarte gespeicherten Daten, sowie Informationen für die, aufgrund der Sicherheitsvorgaben der Anwendung 10 eventuell erforderliche, Behandlung dieser Daten. Zur einfachen Identifizierung der jeweiligen Daten für einen Zugriff durch die Anwendung 10 werden den Daten ein oder mehrere Alias-Namen zugeordnet. Diese Informationen werden in geeigneter Form, wie z. B. tabellarisch oder hierarchisch, in dem Wörterbuch 30 angelegt und umfassen alle erforderlichen Informationen für die Anwendung 10 oder eine Vielzahl weiterer Anwendungen (näheres dazu siehe Fig. 2). Der Zugriff auf das Wörterbuch 30 erfolgt nur durch den Agenten 40. Die Zuordnung eines Wörterbuchs 30 zu einem Agenten 40 kommt durch den externen Anwendungsteil oder einer speziellen Erweiterung des Agenten 40 zustande.

Die Generierung des Wörterbuches 30 geschieht vorzugsweise durch eine manuelle Erstellung der erforderlichen Daten. Dazu werden sämtliche Informationen für alle auf der Chipkarte 20 befindlichen Daten und deren Eigenschaften benötigt. Die von der Anwendung 10 auf der Chipkarte 20 erreichbaren Daten werden mit den für das Wörterbuch 30 vorgesehenen Alias-Namen versehen.

Sollen Daten der Chipkarte 20 prozessiert werden, stellt die Anwendung 10 eine Anfrage 100 an den Agenten 40. Eine solche Anfrage 100 enthält Informationen über die gewünschte Zugriffsmethode, sowie den Alias-Namen der gewünschten Daten. Der Agent 40 erzeugt auf diese Anfrage 100 hin die erforderlichen Kommandosequenzen, um auf Daten der Chipkarte 20 zugreifen zu können. Er bedient sich dabei der Informationen, die in dem Wörterbuch 30 in einem Eintrag für den jeweiligen Alias-Namen gespeichert sind. Die erforderlichen Parameter der Zugriffsmethode werden mit der Anfrage 100 an den Agenten 40 übergeben. Beispielsweise gehören zu der Anfrage 100, die ein Lesen eines Datums auf der Chipkarte 20 bewirken soll, der Alias-Name des Datums, die zu verwendende Lesemethode und Angaben über den Ort zur Hinterlegung der Antwort auf die

Anfrage 100. Eine solche Anfrage kann also zum Beispiel aus der Zugriffsart LESEN für Daten mit dem Alias-Namen KONTONUMMER und einer Speicheradresse zur Hinterlegung der Daten bestehen. Neben den Methoden Lesen und Schreiben können hier u. a. auch Authentisierung, Erzeugen oder Löschen von Strukturen, sowie die bei Chipkarten realisierbaren anwendungsspezifischen Kommandos auftreten. Bei diesen handelt es sich um auf der Chipkarte 20 gespeicherte Methoden die durch spezielle Kommandos ausgelöst werden. Dies kann zum Beispiel die Modifikation des Betrags einer auf dem Chip der Chipkarte 20 implementierten Geldbörse sein.

Um eine leichte Einbettung in existierende Systeme zu ermöglichen, ist es vielfach erforderlich, eine asynchrone Funktionsweise des Agenten 40 zu garantieren. Viele der verwendeten Softwareplattformen erfordern eine ereignisgesteuerte Programmierung. Ein synchron arbeitendes Modul kann eventuell zu einer temporären Blockierung des gesamten Systems führen. Die Dauer dieser Blockierung ist für Chipkarten-Anwendungen jedoch vielfach nicht tragbar, da diese eine Datenübertragung zu einem relativ langsamen Medium, sowie eine Bearbeitung von Kommandos auf einem relativ langsamen Prozessor einschließt.

Der Agent 40 wird vorzugsweise als asynchrones Modul implementiert, läßt sich jedoch auch synchron betreiben. Auf die Anfrage 100 der Anwendung 10 hin, generiert der Agent 40 eine Sequenz von Kommandos für die Chipkarte. Ein jedes Kommando 110 der Sequenz wird einzeln an die Anwendung 10 zurückgegeben. Diese überträgt das jeweilige Kommando 110 des Agenten 40 als ein Anwendungskommando 130 an ein entsprechendes Lese-/Schreibgerät 120 zum unmittelbaren Lesen und Schreiben von Daten auf der Chipkarte 20. An dieser Stelle sei erwähnt, daß die Anwendung 10, wie bereits oben beschrieben wurde, auch direkt in dem Lese-/Schreibgerät 120 lokalisiert sein kann. Auch kann für bestimmte Anwendung anstatt einer Sequenz von Kommandos ein einzelnes Kommando 110 ausreichen.

Das Lese-/Schreibgerät 120 überträgt das Anwendungskommando 130, oder ein davon abgeleitetes Anwendungskommando 130', auf die Chipkarte 20 und empfängt von diesem eine Antwort 140. Die Daten der Antwort 140, oder einer ebenfalls davon abgeleitete Antwort 140', werden von der Anwendung 10 entgegengenommen und wiederum als Antwortdaten 150 dem Agenten 40 zugeführt. Dieser interpretiert die Antwortdaten 150 und generiert daraufhin, falls erforderlich, das nächste Kommando. Der Vorgang wird solange wiederholt, bis die gewünschten Daten prozessiert worden sind. Am Ende des Vorganges, oder auch kontinuierlich während der Kommandosequenz, übermittelt der Agent 40 einen Datensatz 160 als Reaktion auf die Anfrage 100 und die von der Chipkarte 20 zurückgegebenen Daten an die Anwendung 10. Dieser Datensatz 160 stellt die, für die Anwendung 10 verständliche Antwort der Chipkarte 20 auf die, für die Chipkarte 20 nicht verständliche Anfrage 100 der Anwendung 10 dar.

Die Realisierung des Agenten 40 geschieht vorzugsweise durch die Verwendung des Konzeptes der endlichen Automaten. Ein solcher Automat ist in der Lage, seinen internen Zustand zu speichern und aufgrund von Eingabedaten und seinem aktuellem Zustand in einen anderen Zustand zu wechseln. Die Programmierung dieses Automaten geschieht durch die Vorgabe der gewünschten Zustandsänderungen für die jeweiligen Kombinationen aus augenblicklichem Zustand und Eingabedaten.

Während der Generierung von Kommandos kommt es bei Verwendung von durch Schlüssel geschützten Daten oder Befehlen, z. B. auf der Chipkarte 20, auch zu den erforderlichen Ver- und Entschlüsselungen. Diese sollen aufgrund der Sicherheitsvorgaben vielfach nicht durch den Agenten 40 selbst durchgeführt werden. Der Agent 40 wird daher durch eine geeignete Anfrage an die Anwendung 10 diesen Bedarf signalisieren und die erforderlichen Daten liefern. Die Anwendung 10 ihrerseits wird in den meisten Fällen diese Anfrage des Agenten 40 an ein spezialisiertes Modul zur Verschlüsselung weiterleiten. Nach der Antwort kann der Agent 40 mit der Generierung der Kommandosequenz fortfahren.

Neben speziellen Anfragen zur Verschlüsselung können unter anderem noch Anfragen an das Lese-/Schreibgerät 120, sowie Anfragen zu einem (nicht gezeigten) PIN Eingabegerät auftreten. Diese Anfragen werden durch die Anwendung 10 an die entsprechenden Geräte zur Bearbeitung weitergeleitet.

Um mit der Chipkarte 20 zu kommunizieren, wird die Anwendung 10 zunächst den geeigneten Agenten 40 (beispielsweise aus einer Vielzahl möglicher Agenten) auswählen. Spätestens bei der Generierung einer Anfrage an den Agenten 40 wird auch das gewünschte Wörterbuch 30 spezifiziert. Beides geschieht aufgrund spezieller Antwortdaten der Karte, die z. B. durch eine anfängliche Abfrage von der Chipkarte 20 erhalten wurden, und Kenntnissen über die jeweilige, auszuführende Anwendung 10. Nach Auswahl des Agenten 40 und des Wörterbuches 30 kann durch die Anwendung 10 die Anfrage 100 an den Agenten 40 gestellt werden. Der Agent 40 durchsucht daraufhin das Wörterbuch 30 nach dem in der Anfrage 100 enthaltenen Alias-Namen und erhält darüber auch die erforderlichen anwendungsspezifischen Informationen. Die erste Anfrage 100 versetzt den Agenten 40 in einen Startzustand und es wird daraufhin der erste Befehl der erforderlichen Kommandosequenz generiert. Mit jeder Antwort auf ein Kommando wird der Agent 40 den internen Zustand wechseln und Informationen über den Fortschritt der Sequenz sammeln. Die Generierung von Kommandos bricht ab, sobald der Agent 40 einen, für die Anfrage relevanten, Endzustand erreicht hat. Dies ist der Fall, wenn ein Fehler bei der Kommunikation aufgetreten sein sollte, oder die Daten erfolgreich prozessiert worden sind.

Sollte eine Änderung der Anwendung 10 durch eine Modifikation der auf Chipkarten gespeicherten Daten erforderlich werden, kann dies auf eine Änderung des Wörterbuchs 30 begrenzt werden. Ein neues Wörterbuch 30' (nicht gezeigt) wird der Anwendung 10 verfügbar gemacht, und die Anwendung 10 selbst muß nicht geändert werden. Eine Änderung der auf die Chipkarte 20 bezogenen Teile der Anwendung 10, wie zum Beispiel die Verwendung anderer Chipkartentypen mit unterschiedlicher Schnittstelle, führt zu einer Generierung eines neuen Agenten 40' (nicht gezeigt). Die datenbezogenen Informationen zu der Anwendung 10 in dem Wörterbuch 30 oder 30' müssen dabei nicht geändert werden.

Es ist zu verstehen, daß das Kommando 110 und die Antwort 140 des Agenten 40 vorzugsweise weder durch die Anwendung 10 noch durch das Lese-/Schreibgerät 120 oder eventuell andere zwischen Agent 40 und

Chipkarte 20 liegenden Geräte verändert wird. Die Signale 110, 130 und 130' sind in diesem Fall identisch, und die Chipkarte 20 erhält mittelbar das Kommando 110 des Agenten 40 und übermittelt diesem wiederum (gleichfalls mittelbar) seine Antwort 140 auf das Kommando 110. Damit sind auch die Signale 140, 140' und 150 identisch. Analog dazu ist auch eine jeweilige unmittelbare Übertragung des Kommandos 110 an die Chipkarte 20 und der entsprechenden Antwort 140 zurück an den Agenten 40 realisierbar.

Es ist auch zu verstehen, daß der Agent 40 sowohl in einem asynchronen als auch in einem synchronen Modus betrieben werden kann. Im asynchronen Modus wartet die Anwendung 10 nicht auf eine Rückmeldung des Agenten 40 auf die Anfrage 100 und steht somit auch zwischen Anfrage 100 und Rückmeldung zur Verfügung. Im synchronen Modus jedoch wartet die Anwendung 10 auf eine Rückmeldung des Agenten 40 auf die Anfrage 100 und steht erst nach erfolgter Rückmeldung wieder zur Verfügung.

Fig. 2 zeigt die Verwendung mehrerer Wörterbücher 230 und/oder mehrerer Agenten 240 für eine Vielzahl verschiedener Anwendungen 210 und Typen von Chipkarten 220. Eine sogenannte Agentur 250 wird durch die Anwendung 10 aus der Vielzahl von Anwendungen 210 anstelle des Agenten 40, wie in Fig. 1, aufgerufen. Aufgabe der Agentur 250 ist es, die verschiedenen Wörterbücher 230 und Agenten 240 zu verwalten und bei Bedarf für die Verfügbarkeit entsprechender Versionen zu sorgen. Dazu wird vorzugsweise eine Liste aller lokalen Wörterbücher 230 und Agenten 240 mit deren Eigenschaften sowie der Kriterien, wann diese zu verwenden sind, benutzt. Ist kein geeigneter Agent oder kein geeignetes Wörterbuch verfügbar, kann von einer anderen Agentur 260 ein benötigtes Exemplar, oder eine Kopie davon, angefordert werden. Dies kann über ein lokales Netzwerk, sowie über geeignete Kommunikationswege zwischen den Agenturen 250 und 260 geschehen.

Wird bei der Implementierung eines Agenten 40 eine interpretierte Programmiersprache verwendet, kann ein und dieselbe Implementierung auch auf Geräten mit unterschiedlicher Hardware und Betriebssystemen verwendet werden. Dazu wird für jedes Gerät ein passender Interpreter implementiert, der das Programm des Agenten interpretierend abarbeitet.

Sind beim Transport von Kommandos vom Agenten 40 zur Chipkarte 20 mehrere Programm-Ebenen und Schnittstellen zu durchlaufen, ergibt sich eventuell ein deutlicher Overhead bei der Kommunikation mit der Chipkarte 20. Dieser kann verringert werden, indem man, wie in Fig. 3 gezeigt, den Agenten 40 näher zur Chipkarte 20 hin plziert. Dazu wird der Agent 40 durch einen Stellvertreter, den sogenannten Proxy 340 ersetzt. Dieser erstellt für jede Anfrage 100 einen einzigen Datensatz 350 mit allen erforderlichen Informationen aus dem Wörterbuch 30 und läßt diesen an einen Router 360 senden. Der Router 360 empfängt den Datensatz 350 des Proxy 340, instruiert nun seinerseits den auf diese Ebene verschobenen Agenten 40. Alle vom Agenten 40 generierten Kommandos 370 werden nun durch den Router 360 an die Chipkarte 20 geschickt und jede Antwort 380 wiederum an den Agenten 40. Ein Resultat 390 der Anfrage 100 nach der Kommunikation des Agenten 40 mit der Chipkarte 20 wird schließlich vom Router 360 an den Proxy 340 und von diesem als Antwort 160 an die Anwendung 10 zurückgegeben. Der Proxy 340 garantiert eine einheitliche Schnittstelle für die Anwendung 10, so daß diese bei seiner Verwendung nicht geändert werden muß. Die Anwendung 10 bleibt unabhängig davon, ob sie mit dem Agenten 40 oder dessen Proxy 340 kommuniziert. Der Router 360 wird vorzugsweise auf der gleichen Ebene wie der Agent 40 plziert, während das Wörterbuch 30 auf der Ebene der Anwendung 10, beziehungsweise der Agentur 230, verbleibt.

Es sei angemerkt, daß das Lese-/Schreibgerät 120 eine abstrakte Darstellung eines Gerätes mit einem unmittelbaren Zugriff auf die Chipkarte 20 darstellt. Das Lese-/Schreibgerät 120 kann jedoch aus einer Vielzahl einzelner Geräte und Schichten mit einzelnen Schnittstellen untereinander bestehen.

Es ist zu verstehen, daß sich die Erfindung sowohl auf die einfachen Speicherkarten als auch auf intelligente Datenträgerkarten mit einem eigenen Prozessor zur Kontrolle der auf dem Chip der Datenträgerkarte gespeicherten Daten bezieht.

Eine Modifikation des Wörterbuches 30 läßt sich erfindungsgemäß auch unmittelbar durch den Agenten 40 bzw. die Agentur 250 bewerkstelligen. Dies ist insbesondere dann von Vorteil, wenn Strukturen auf der Chipkarte 20 mit Hilfe des Agenten 40 erzeugt oder verändert werden sollen, z. B. wenn der Agent 40 eine neue Datei auf der Chipkarte 20 neu anlegen oder umgestalten soll. Diese neue Information wird dann von dem Agenten 40 bzw. der Agentur 250 in das Wörterbuch 30 eingetragen.

In einer Ausführungsform der Erfindung befindet sich das Wörterbuch 30 unmittelbar auf der Chipkarte 20. Dies ermöglicht, daß immer das benötigte Wörterbuch 30 für die Chipkarte 20 unmittelbar zur Verfügung steht. In einer weiteren Ausführungsform wird von der Chipkarte 20 das unmittelbar auf dieser Chipkarte 20 sich befindliche Wörterbuch 30 kopiert und gespeichert (vorzugsweise durch den Agenten 40) und steht dem Agenten 40 bzw. der Agentur 250 für weitere Chipkarten desselben Chipkartentypes zur Verfügung.

#### Detailliertes Ausführungsbeispiel

Das folgende Beispiel soll den Ablauf einer Kommandosequenz verdeutlichen. Die Anwendung 10 möchte eine Kontonummer von der Chipkarte 20 lesen. Der dabei zu verwendende Agent 40 wird anhand von Kartenmerkmalen des Chipkartentypes der Chipkarte 20 selektiert. Die Anwendung 10 schickt die Anfrage 100 an den Agenten 40, dem bereits ein spezifisches Wörterbuch 30 durch eine vorangegangene Auswahl anhand von Kartendaten des Chipkartentypes der Chipkarte 20 zugeordnet sein soll. Tabelle 1 zeigt einen Ausschnitt aus dem zugeordneten Wörterbuch 30.

Die Anfrage 100 enthält als Parametern den Befehl "LESEN", den Aliasnamen "KONTONUMMER" und die Speicheradresse eines Puffers, in dem das Ergebnis auf die Anfrage 100 abgelegt werden soll. Der Agent 40 durchsucht darauf hin das Wörterbuch 30 nach dem Aliasnamen "KONTONUMMER" (siehe Tabelle 1). Der gefundene Eintrag liefert die Informationen, in welchem Verzeichnis und in welcher Datei der Chipkarte 20 dieses Datum zu finden ist, sowie Angaben über die Art und den Umfang des Datums und sicherheitsrelevante

Angaben. Die Daten auf der Chipkarte 20 seien in diesem Beispiel in Verzeichnissen angeordnet, wie diese von Dateisystemen für Computer bekannt sind.

In diesem Beispiel wird in Tabelle 1 (mit Pfeil markiert) unter dem Aliasnamen "KONTONUMMER" die Pfadangabe "3F00.A100.4001" gefunden und die Angabe, das es sich um eine Datei des Typs TRANSPARENT handelt. Das Datum befindet sich in der Datei mit Offset 4 und einer Länge von 5 bytes. Zur Übersetzung der logischen Schlüsselnummern in physikalische Schlüssel dient die Angabe der Schlüsselldomäne KREDIT und der Autorisierungsdomäne AUT KREDIT. Bei der Eingabe von Paßwörtern (mit einer PIN) wird für PIN1 und PIN2 die Domäne GLOBAL verwendet. Eine Domäne für ein bestimmtes Objekt stellt eine Menge von Verzeichnissen und Dateien dar, die auf eine gemeinsame Instanz dieses Objektes zugreifen.

Der Agent 40 beginnt mit der Generierung einer Kommandosequenz zur Selektion des erforderlichen Verzeichnisses auf der Chipkarte 20 anhand der Pfadangabe "3F00.A100.4001". Dabei wird die augenblickliche Verzeichnisselektion in Betracht gezogen. Für den Fall, daß das Verzeichnis 3F00 bereits selektiert ist, wird als ein erstes Kommando 110 zur Selektion des Unterverzeichnisses A100 generiert. Dieses erste Kommando 110 wird an die Anwendung 10 zurückgegeben mit einem Signal, das die Kommandosequenz fortgesetzt werden soll. Die Anwendung 10 gibt das erste Kommando 110 als Kommando 130 daraufhin an eine Schnittstelle zur Übertragung an das verwendete Schreib/Lesegerät 120 für die Chipkarte 20 weiter. Dieses überträgt das Kommando 130 an die Chipkarte 20. Die Chipkarte 20 sendet eine Antwort 140 mit einer Bestätigung der Auswahl und gegebenenfalls Informationen über das selektierte Verzeichnis. Die Antwort 140 wird vom Schreib/Lesegerät 120 an die Anwendung 10 geschickt, die diese sofort als Antwortdaten 150 an den Agenten 40 weiterleitet.

Die Antwortdaten 150 liegen im Allgemeinen in einem der Anwendung 10 unverständlichen Format vor. Um die Antwortdaten verstehen zu können, müssen die umfangreichen Kenntnisse des Agenten 40 über die Chipkarte 20 vorhanden sein. Der Agent 40 kann aus den Antwortdaten 150 Schlüsse über den Erfolg des ersten Kommandos 110 ziehen und Informationen über den Fortschritt der Kommandosequenz sammeln. Als nächstes wird ein zweites Kommando 110' zur Selektion der Datei 4001 an die Anwendung 10 übermittelt und wie zuvor das erste Kommando 110 behandelt.

Anschließend wird ein weiteres Kommando 110'' zum Lesen der angeforderten Daten generiert. Dazu wird die Information über Ort (Offset=4) und Größe (Länge=5) der Kontonummer verwendet. Das Kommando 110'' wird wie zuvor Kommando 110 übertragen und entsprechend von der Chipkarte 20 beantwortet. Die Antwortdaten 150'' der Chipkarte 20 enthalten nun außerdem die gewünschten Daten. Diese Daten seien beispielsweise verschlüsselt, was der Agent 40 anhand vorhergehender Informationen bei der Selektion der Datei erkannt hat. Nun wird eine Anfrage 160 des Agenten 40 an die Anwendung 10 generiert, um eine Entschlüsselung der Daten durchzuführen. Die Anwendung 10 kann die Entschlüsselung selbst durchführen oder sie an ein weiteres spezialisiertes Modul weiterleiten. Das Ergebnis wird als Antwortdatum 150''' an den Agenten 40 zurückgegeben. Dieser kopiert nun die Daten in einem der Anwendung 10 verständlichen Format in den bei der ursprünglichen Anfrage 100 angegebenen Pufferspeicher und signalisiert der Anwendung 10 das Ende der Kommandosequenz. Die Anwendung 10 kann nun auf die Kontonummer zugreifen und eine nächste Anfrage an den Agenten 40 senden.

Tabelle 1

Alias Name	Pfadangabe	Datentyp Modus	Offset Index	Länge Domäne	Key Domäne	Autorisier Domäne	Pin 1 Domäne	Pin 2 Domäne
"BANKLEITZAHL"	"3F00.A100.4001"	TRANSPARENT	0,	4,	KREDIT,	AUT_KREDIT,	GLOBAL,	GLOBAL
"KONTONUMMER"	"3F00.A100.4001"	TRANSPARENT	4,	5,	KREDIT,	AUT_KREDIT,	GLOBAL,	GLOBAL
"KARTENNUMMER"	"3F00.A100.4001"	TRANSPARENT	9,	10,	KREDIT,	AUT_KREDIT,	GLOBAL,	GLOBAL
"DEBIT"	"3F00.A200"	DEDICATED	0,	0,	DEBIT,	AUT_DEBIT,	GLOBAL,	GLOBAL
"DEBIT AMOUNT"	"3F00.A200.1200"	CYCLIC_FIXED	2,	0,	DEBIT,	AUT_DEBIT,	GLOBAL,	GLOBAL

## Patentansprüche

1. Vorrichtung zur Kommunikation einer Anwendung (10) mit einer Chipkarte (20), gekennzeichnet durch:  
mindestens ein Anwendungsdatenverzeichnis (30) zur Aufnahme von Informationen über anwendungsspe-  
zifische Daten der Anwendung (10), und  
mindestens ein Chipkartendialogmodul (40) für die Generierung von Kommandos mit Hilfe des Anwen-  
dungsdatenverzeichnisses (30) für eine Schnittstelle zur Chipkarte (20), wobei das Chipkartendialogmodul  
(40) kartenspezifische Daten über die Chipkarte (20) enthält.

2. Vorrichtung nach Anspruch 1, dadurch gekennzeichnet, daß das Anwendungsdatenverzeichnis (30) Alias-Namen zur Bezeichnung der anwendungsspezifischen Daten enthält.
3. Vorrichtung nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß nur das Chipkartendialogmodul (40) Zugriff auf das Anwendungsdatenverzeichnis (30) hat.
- 5 4. Vorrichtung nach einem der vorstehenden Ansprüche, gekennzeichnet durch ein Zuordnungsmittel zur Zuordnung eines Anwendungsdatenverzeichnisses (30) zu einem Chipkartendialogmodul (40).
5. Vorrichtung nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß in dem Anwendungsdatenverzeichnis (30) die Informationen für eine oder mehrere Anwendungen (10) enthalten sind.
- 10 6. Vorrichtung nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß das Chipkartendialogmodul (40) so implementiert ist, daß es sowohl als synchrones als auch als asynchrones Modul verwendet werden kann.
7. Vorrichtung nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß das Chipkartendialogmodul (40) als ein endlicher Automat implementiert ist, der in der Lage ist, sich seinen internen Zustand zu merken und aufgrund von Eingabedaten und aktuellem Zustand in einen anderen Zustand zu wechseln.
- 15 8. Vorrichtung nach einem der vorstehenden Ansprüche, gekennzeichnet durch mindestens eine Agentur (250, 260), die für die Verwaltung der jeweils vorhandenen Chipkartendialogmodule (240, 240') und Anwendungsdatenverzeichnisse (230, 230') zuständig ist und von der Anwendung (10) benutzt wird, um die entsprechende Kombination von Chipkartendialogmodul (40) und Anwendungsdatenverzeichnis (30) für den jeweiligen Typ von Chipkarten (20) zu bestimmen.
- 20 9. Vorrichtung nach Anspruch 8, dadurch gekennzeichnet, daß die mindestens eine Agentur (250) mit einer weiteren Agentur (260) über ein Netzwerk oder einen anderen, geeigneten Kommunikationsweg in Verbindung steht.
10. Vorrichtung nach einem der vorstehenden Ansprüche, gekennzeichnet durch einen Stellvertreter (340), der in etwa in der Ebene der Anwendung (10) platziert ist und das Chipkartendialogmodul (40) in seiner Kommunikation mit der Anwendung (10) ersetzt, der Stellvertreter (340) weist ein Mittel zur Erstellung eines Datensatzes (350) auf, wobei der Datensatz (350) auf eine Anfrage (100) der Anwendung (10) hin Informationen aus dem Anwendungsdatenverzeichnis (30) enthält;
- 25 einen Router (360) mit einem Mittel zum Empfang des Datensatzes (350) und einem Mittel zur Instruktion des auf die Ebene des Router (360) verschobenen Chipkartendialogmoduls (40),
- 30 wobei der Router (360) ein von dem Chipkartendialogmodul (40) generiertes Kommando (370) an die Chipkarte (20) und eine Antwort (380) der Chipkarte (20) auf das Kommando (370) an das Chipkartendialogmodul (40) schickt, der Router (360) ein Resultat (390) der Anfrage (100) nach der Kommunikation des Chipkartendialogmoduls (40) mit der Chipkarte (20) an den Stellvertreter (340) zurückgibt, und der Stellvertreter (340) eine Antwort (160) an die Anwendung (10) gibt.
- 35 11. Verfahren zur Kommunikation einer Anwendung (10) mit einer Chipkarte (20), dadurch gekennzeichnet, daß von der Anwendung (10) eine Anfrage (100) zur Kommunikation mit der Chipkarte (20) an ein für die Chipkarte (20) spezifiziertes Chipkartendialogmodul (40) gestellt wird, das kartenspezifische Daten über die Chipkarte (20) aufweist, und
- 40 das Chipkartendialogmodul (40) auf die Anfrage (100) hin mindestens ein Kommando (110) erzeugt, das zu einer Kommunikation mit der Chipkarte (20) erforderlich ist, wobei das Chipkartendialogmodul (40) sich dabei anwendungsspezifische Informationen, die in einem Anwendungsdatenverzeichnis (30) gespeichert sind, zugänglich macht.
12. Verfahren nach Anspruch 11, dadurch gekennzeichnet, daß
- 45 das Chipkartendialogmodul (40) durch die Anwendung (10) aus einer Vielzahl von Chipkartendialogmodulen (40) ausgewählt wird;
- spätestens bei der Generierung der Anfrage (100) an das Chipkartendialogmodul (40) das Anwendungsdatenverzeichnis (30) spezifiziert wird;
- wobei beides aufgrund spezieller Antwortdaten der Chipkarte (20) und Kenntnissen über die jeweilige, auszuführende Anwendung (10) geschieht.
- 50 13. Verfahren nach Anspruch 12, dadurch gekennzeichnet, daß die speziellen Antwortdaten der Chipkarte (20) durch eine anfängliche Abfrage von der Chipkarte (20) erhalten wurden.
14. Verfahren nach einem der Ansprüche 11—13, dadurch gekennzeichnet, daß in dem Anwendungsdatenverzeichnis (30) zur Identifizierung den jeweiligen Daten für einen Zugriff durch die Anwendung (10) mindestens ein Alias-Name zugeordnet wird.
- 55 15. Verfahren nach Anspruch 14, dadurch gekennzeichnet, daß die Anfrage (100) Informationen über die gewünschte Zugriffsmethode sowie den Alias-Namen der gewünschten Daten enthält.
16. Verfahren nach Anspruch 15, dadurch gekennzeichnet, daß nach Stellen der Anfrage (100) an das Chipkartendialogmodul (40) dieses das Anwendungsdatenverzeichnis (30) nach in der Anfrage 100 enthaltenen Alias-Namen durchsucht.
- 60 17. Verfahren nach einem der Ansprüche 11—16, dadurch gekennzeichnet, daß die erste Anfrage (100) das Chipkartendialogmodul (40) in einen Startzustand versetzt, daraufhin ein erstes Kommando einer Kommandosequenz generiert wird, wobei mit jeder Antwort (150) auf ein Kommando (110) das Chipkartendialogmodul (40) den internen Zustand wechselt und Informationen über den Fortschritt der Sequenz sammelt, und die Generierung von Kommandos abbricht, sobald das Chipkartendialogmodul (40) einen, für die Anfrage (100) relevanten, Endzustand erreicht hat.
- 65 18. Verfahren nach Anspruch 17, dadurch gekennzeichnet, daß der Endzustand erreicht wird, wenn ein Fehler bei der Kommunikation aufgetreten sein sollte oder die Daten erfolgreich prozessiert worden sind.



19. Verfahren nach einem der Ansprüche 11–18, dadurch gekennzeichnet, daß bei einer Modifikation der auf der Chipkarte (20) gespeicherten Daten nur eine Änderung des jeweiligen Anwendungsdatenverzeichnis (30) durchgeführt und ein neues Anwendungsdatenverzeichnis (30') der Anwendung (10) verfügbar gemacht wird, wobei die Anwendung (10) selbst nicht geändert werden muß.
20. Verfahren nach einem der Ansprüche 11–19, dadurch gekennzeichnet, daß bei einer Änderung der auf die Chipkarte (20) bezogenen Teile der Anwendung (10) eine Generierung eines neuen Chipkartendialogmoduls (40') durchgeführt wird.
21. Verfahren nach einem der Ansprüche 11–20, dadurch gekennzeichnet, daß bei der Verwendung mehrerer Anwendungsdatenverzeichnis (230) und/oder mehrerer Chipkartendialogmodule (240) für eine Vielzahl verschiedener Anwendungen (210) und Typen von Chipkarten (220) eine Agentur (250) durch die Anwendung (10) aufgerufen wird, wobei die Agentur (250) die verschiedenen Anwendungsdatenverzeichnisse (230) und Chipkartendialogmodule (240) verwaltet.
22. Verfahren nach Anspruch 21, dadurch gekennzeichnet, daß die Agentur (250) bei Bedarf für die Verfügbarkeit entsprechender Versionen von Anwendungsdatenverzeichnissen (230) und Chipkartendialogmodulen (240) sorgt.
23. Verfahren nach Anspruch 20 oder 21, dadurch gekennzeichnet, daß die Agentur (250) von einer anderen Agentur (260) ein benötigtes Exemplar entsprechender Versionen von Anwendungsdatenverzeichnissen (230) und/oder Chipkartendialogmodulen (240), oder eine Kopie davon, anfordert.
24. Verfahren nach einem der Ansprüche 20–23, dadurch gekennzeichnet, daß die Agentur (250) fehlende Teile durch Kommunikation mit anderen, lokal oder über Vernetzung erreichbaren Ressourcen, beschafft.
25. Verfahren nach einem der Ansprüche 11–24, dadurch gekennzeichnet, daß das erzeugte Kommando (110) an die Anwendung (10) zurückgegeben wird und von dieser als ein Anwendungskommando (130) an ein entsprechendes Lese-/Schreibgerät (120) zum unmittelbaren Lesen und Schreiben von Daten auf der Chipkarte (20) weitergegeben wird; das Lese-/Schreibgerät (120) das Anwendungskommando (130) oder ein davon abgeleitetes Anwendungskommando (130') auf die Chipkarte (20) überträgt und von dieser eine Antwort (140) empfängt; die Antwort (140), oder eine ebenfalls davon abgeleitete Antwort (140'), von der Anwendung (10) entgegengenommen wird und wiederum als Antwortdaten (150) dem Chipkartendialogmodul (40) zugeführt werden; das Chipkartendialogmodul (40) die Antwortdaten (150) interpretiert und daraufhin ein nächstes Kommando generiert; wobei der Vorgang solange wiederholt wird, bis die Anfrage (100) prozessiert worden sind.
26. Verfahren nach Anspruch 25, dadurch gekennzeichnet, daß am Ende des Vorganges, oder auch kontinuierlich während einer Kommandosequenz, das Chipkartendialogmodul (40) einen Datensatz (160) als Reaktion auf die Anfrage (100) und die von der Chipkarte (20) zurückgegebenen Daten an die Anwendung (10) übermittelt, wobei der Datensatz (160) die, für die Anwendung (10) verständliche Antwort der Chipkarte (20) auf die, für die Chipkarte (20) nicht verständliche Anfrage (100) der Anwendung (10) darstellt.
27. Verfahren nach einem der Ansprüche 11–26, dadurch gekennzeichnet, daß bei Verwendung von durch Schlüssel geschützten Daten oder Befehlen, das Chipkartendialogmodul (40) mit einer geeigneten Anfrage an die Anwendung (10) einen Bedarf zur Durchführung der erforderlichen Ver- und Entschlüsselungen signalisiert und die erforderlichen Daten an diese liefert; und die Anwendung (10) die Anfrage des Chipkartendialogmoduls (40) an ein spezialisiertes Modul zur Verschlüsselung weiterleitet.
28. Verfahren nach einem der Ansprüche 11–27, dadurch gekennzeichnet, daß für die Prozessierung von Daten und/oder Programmen auf der Chipkarte (20) durch die Anwendung (10) die Angabe eines Alias-Namens eines Datums und/oder eines Programmes und eine gewünschte Zugriffsmethode übergeben wird.
29. Verfahren nach Anspruch 28, dadurch gekennzeichnet, daß als Zugriffsmethoden Lesen, Schreiben, Authentisierung, und/oder Erzeugen oder Löschen von Strukturen auftreten können.
30. Verfahren nach Anspruch 27 oder 28, dadurch gekennzeichnet, daß als Zugriffsmethode die bei der Chipkarte (20) realisierbaren anwendungsspezifischen Kommandos auftreten können, die auf der Chipkarte (20) gespeichert und durch spezielle Kommandos ausgelöst werden können.
31. Verfahren nach einem der Ansprüche 11–30, dadurch gekennzeichnet, daß ein Stellvertreter (340), der in etwa in der Ebene der Anwendung (10) plaziert ist und das Chipkartendialogmodul (40) in seiner Kommunikation mit der Anwendung (10) ersetzt, auf die Anfrage (100) der Anwendung (10) hin einen Datensatz (350) erstellt, der Informationen aus dem Anwendungsdatenverzeichnis (30) enthält; ein Router (360) den Datensatz (350) empfängt und das, auf diese Ebene des Router (360) verschobene, Chipkartendialogmodul (40), instruiert; der Router (360) ein von dem Chipkartendialogmodul (40) generiertes Kommando (370) an die Chipkarte (20) und eine Antwort (380) der Chipkarte (20) auf das Kommando (370) an das Chipkartendialogmodul (40) schickt, der Router (360) ein Resultat (390) der Anfrage (100) nach der Kommunikation des Chipkartendialogmoduls (40) mit der Chipkarte (20) an den Stellvertreter (340) zurückgibt, und der Stellvertreter (340) eine Antwort (160) an die Anwendung (10) gibt.
32. Verfahren nach einem der Ansprüche 11–31, dadurch gekennzeichnet, daß die Generierung des Anwendungsdatenverzeichnisses (30) durch eine manuelle Erstellung der erforderlichen Daten geschieht, wobei die Generierung mit Hilfe sämtlicher Informationen für alle auf der Chipkarte (20) befindlichen Daten und deren Eigenschaften durchgeführt wird und die von der Anwendung (10) auf der Chipkarte (20) erreichbaren Daten mit Alias-Namen versehen werden.

33. Verfahren nach einem der Ansprüche 11–32, dadurch gekennzeichnet, daß eine Modifikation des Anwendungsdatenverzeichnisses (30) unmittelbar durch das Chipkartendialogmodul (40) oder die Agentur (250) bewerkstelligt wird.

34. Verfahren nach einem der Ansprüche 11–33, dadurch gekennzeichnet, daß ein auf der Chipkarte (20) sich befindliches Anwendungsdatenverzeichnis (30) kopiert, gespeichert und dem Chipkartendialogmodul (40) oder der Agentur (250) zur Verfügung gestellt wird.

35. Vorrichtung und/oder Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die anwendungsspezifischen Daten Informationen über Art, Lokalität, Umfang und Zugriffsmethoden für auf einer Chipkarte gespeicherte Daten, sowie über auf der Chipkarte (20) gespeicherten Daten aufweisen.

36. Vorrichtung und/oder Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die kartenspezifische Daten Informationen über die Kommandos und das Protokoll der Chipkarte (20) zum Zugriff auf die dort gespeicherten Daten aufweisen.

37. Vorrichtung und/oder Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die kartenspezifischen Informationen komplette Grundstrukturen der Chipkarte (20) oder für Teile davon enthalten.

38. Vorrichtung und/oder Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die kartenspezifischen Informationen in tabellarischer und/oder hierarchischer Form angelegt sind.

39. Vorrichtung und/oder Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die Chipkarte (20) entweder eine Speicherkarte mit oder ohne einem eigenen Prozessor ist.

40. Vorrichtung und/oder Verfahren nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß sich das Anwendungsdatenverzeichnis (30) auf der Chipkarte (20) befindet.

41. Lese/Schreibgerät für Chipkarten (20), gekennzeichnet durch eine Vorrichtung und/oder Verfahren entsprechend einem der vorstehenden Ansprüche.

42. Verwendung des Verfahrens und/oder der Vorrichtung entsprechend einem der vorstehenden Ansprüche in einem Lese/Schreibgerät für Chipkarten (20).

---

Hierzu 2 Seite(n) Zeichnungen

---

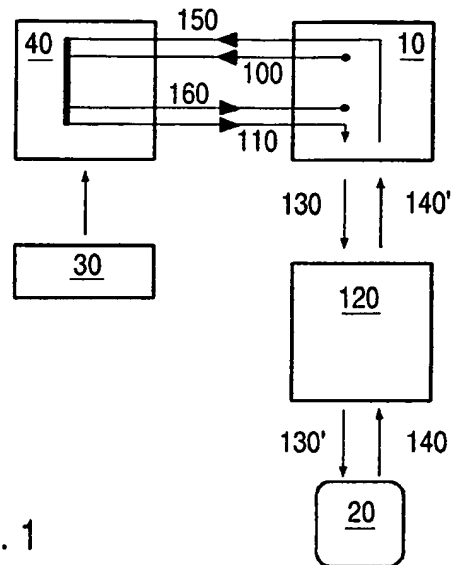


FIG. 1

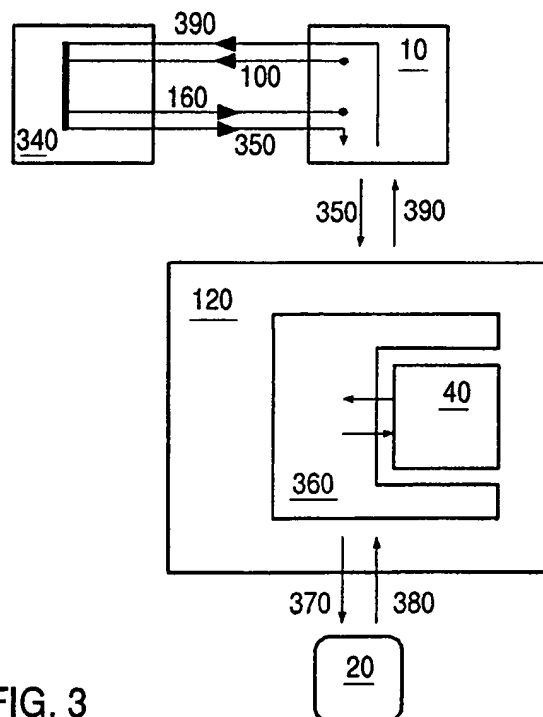


FIG. 3

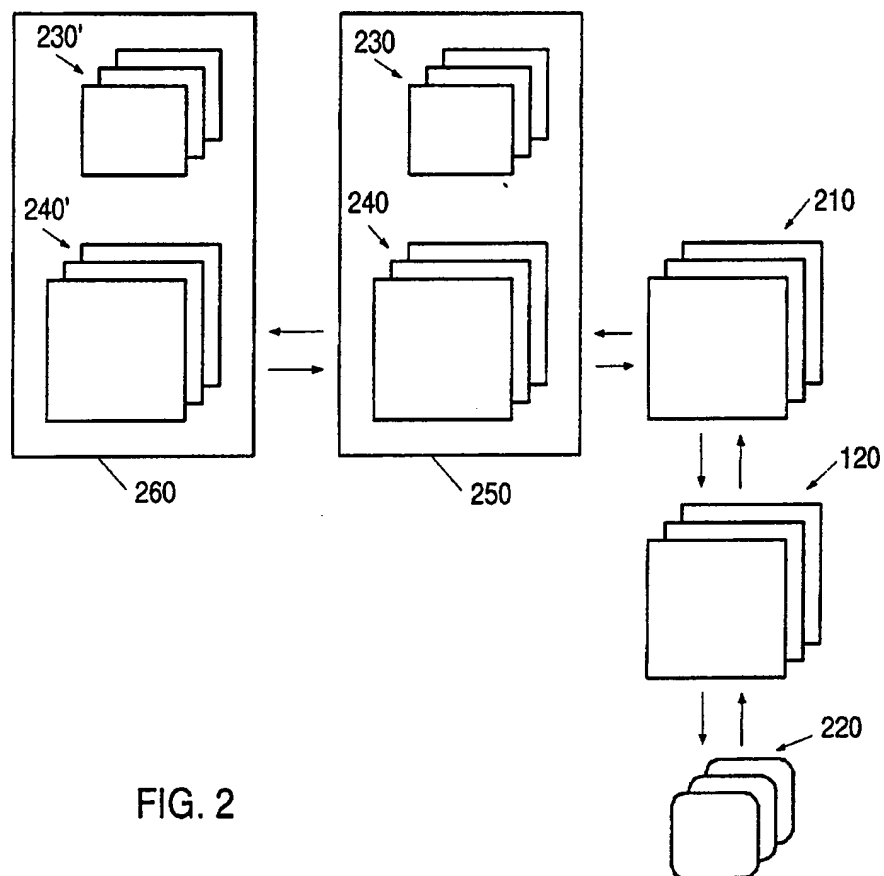


FIG. 2